

Department of Computer Science

Assessed Coursework Assignment Brief

Module code: CS2JA16

Lecturer responsible: Timothy Threadgold

Coursework description: Major Coursework 2: Android Game

Work to be submitted on-line via Blackboard by 12:00 Mid-day on 19th April 2018

Work will be marked and feedback returned by: 10th May 2018

NOTES:

This coursework should be submitted as a **pdf document for the report and a zip file containing your source code and compiled robots.**

By submitting this work you are certifying that it is all your own work and that use of material from other sources has been properly and fully acknowledged in the text. You are also confirming that you have read and understood the University's Statement of Academic Misconduct, available on the University web-pages.

If your work is submitted after the deadline, *10%* of the maximum possible mark will be deducted for *each* working day (or part of) it is late. A mark of zero will be awarded if your work is submitted more than 5 working days late. You are strongly recommended to submit work by the deadline as a late submission on one piece of work can impact on other work.

If you believe that you have a valid reason for failing to meet a deadline then you should complete an Extenuating Circumstances form and submit it to the Student Information Centre *before* the deadline, or as soon as is practicable afterwards, explaining why.

MARKING CRITERIA

a) The table below shows what is typically expected of the work to obtain a given mark.

Classification Range	First Class ($\geq 70\%$)	Upper Second (60..70)	Lower Second (50..60)	Third (40..50)	Pass (30..40)
Section					
Demonstration	Based purely on the Demonstration mark sheet – please look at that for details				
Report	<p>Strong technical knowledge and skill shown through development, proving a strong grasp of object orientation and advanced programming.</p> <p>Report is well structured and fluently written.</p> <p>Design choices are validated in the report, and the work is showing study and implementation of techniques beyond the standard material.</p>	<p>A solid grasp of the subject with a good selection of advanced programming methods.</p> <p>The report is written well, and validates design choices.</p> <p>May show some elements of creativity and originality, and makes use of existing literature to validate choices.</p>	<p>A reasonable range of grasp of the subject, with few technical errors and written in plain English. On topic, relevant, and relatively well organised.</p>	<p>Evidence of appropriate study showing success in progress towards providing a solution with most technical content correct.</p> <p>The work relies on simple examples or uses methods inconsistently.</p>	<p>Shows some evidence of study, but may be largely unfinished, flawed, or irrelevant, whilst showing some attempt to present a coherent solution.</p>

ASSIGNMENT DETAILS

Definition:

In this coursework, you are required to implement an Android Game, and produce an APK from it that can be installed on an Android phone. You will then write a report that describes the design and justifies your code and its design. You can choose the theme of the game.

Problem:

The OOP Design

You are required to develop an OOP design before starting implementation. This should be discussed and submitted in your practical session in week 8 of spring term. This will be marked as that week's practical quiz. You must include an electronic version (e.g. scan handwritten work) in an appendix of the submitted report.

You can later change the design. You must however, justify your changes in the final report.

Your game should:

- Have functionality to **start** individual games on a menu screen using a separate welcome or start screen. The welcome screen must function using the Android API libraries
- Display the current **score** while playing. You define scoring mechanisms
- Have a '**Character**' that the user controls.
 - Controlled by touch on the screen and/or other sensors.
- Have At least three game **levels**
 - Higher marks available for levels stored in files, online, for creating standardised data structures, and for randomly generated levels
 - The levels must have different layouts or opponents with different behaviours
 - The player must have to move in at least two dimensions(i.e. x and y)
- Have **Opponents**
 - Must have more than one opponent on screen at some point in the game play.
 - This can be opponents that the player is attempting to defeat, dodge for points, or compete with
 - The opponents must 'play' the game autonomously, this could be as simple as making random movements for example
- Research how to and implement optimisations to the **memory, and speed** of the game. This process, including tests to verify claims (i.e. you must show the difference in memory usage or speed with and without the optimisations), must be described and justified in the report
- There are 20% marks available for **improvements** (10% each) that extend the code or implement additional features in the system (not gameplay features.) These can take many forms but some of the suggested ways to achieve these marks are below:
 - Creating an online high score list.
 - Researching and improving multithreading of the game by using multiple threads to handle different systems in the game. This should be described and justified in the report
 - Creating a system for the users to create their own levels in game



- o Another improvement that you have designed on top of the core requirements (this should be approved before implementation as it should be the same level of difficulty as one of the above)
- The game should be implemented using the Android libraries and not a game engine (eg Unity, libgdx etc.)

INSTRUCTIONS

YOU SHOULD ENSURE YOU CHANGE THE DEFAULT PACKAGE NAME OF THE CLASSES IN THE PROJECT – OTHERWISE IT MAY NOT INSTALL DURING TESTING

1. **Design:**
 - a. You must create an OOP design in the practical of **Week 8**, which should be submitted online in place of code for the week
 - b. The weekly blackboard quiz will also be related to this and should be answered in full
2. **Demonstration:**
 - a. You are required to demonstrate the final Android game app and to get a demo mark sheet signed by a lecturer or lab assistant. If you have an officially approved extenuating circumstance, then contact Timothy Threadgold by e-mail (*timothy.threadgold@reading.ac.uk*) to set up an alternative demonstration timeslot.
 - b. **Please refer to the demo sheet at the end of this document for the full details of what will be marked in this demonstration. Each of the tick boxes on the form represents one mark unless otherwise specified.**
3. **Report:**
 - a. The report must follow the specified format (see below)
 - b. See the mark breakdown on the next page for the marks given for each section.
4. **Electronic submission**
 - a. On Blackboard (there will be separate submission points for these):
 - i. You are required to submit an electronic **report in pdf format** on Blackboard by the specified deadline.
 - ii. You are required to submit a zip file, including instructions (if required) and the installable APK on Blackboard by the specified deadline.
5. **Gitlab**
 - a. A repository will be created on your csgitlab account that should be used, and committed to every time you make changes to your project.
 - b. This will be used for plagiarism detection, and in case the source code needs to be referred to while marking

MARK BREAKDOWN

Section	Percentage of grade
Demonstration – See demonstration mark sheet for breakdown of marks in this scheme. Each tick box is worth one mark unless otherwise specified.	40
Report– Abstract A 100-300 word overview of the entire project and important features/results	-
Report– Introduction and Showcase A short informal presentation of the application (screenshots with description) in <u>max 2 A4 pages</u> .	10
Report– OOP design A design document of the app (marks are also given for described unimplemented and/or unfinished features) <u>max 4 A4 pages</u> . Include: <ul style="list-style-type: none"> • OOP design with a critical description of how the design aided you in the implementation. If you changed the initial design produced in week 8 you must justify why this was necessary. • A description of special requirements due to designing the application for an Android mobile phone. Make sure to relate this to design and implementation choices in your game. 	20
Report – Memory usage and Speed improvements a description of your research and improvements of memory and speed efficiency of your game <u>max 2 A4 page</u>	10
Report – Improvements/extensions If you researched and implemented and improvement above the other specifications then add a description of the work <u>max 2 A4 pages per improvement made</u>	20 (Max marks available, 10 per improvement)
Report– Conclusions A summary of the important results and conclusions from throughout the rest of the document.	-
Report – Appendices Any extra material as required	-

Coursework #2: DEMONSTRATION MARK FORM

Your full name (PRINT):		DEMO MARK :	
Each tick-box represents one mark unless otherwise specified. Marks will be awarded on the spot.			Mark range
Code style: <ul style="list-style-type: none"> <input type="checkbox"/> Following Java code conventions for all the project, <ul style="list-style-type: none"> <input type="checkbox"/> variable and class names <input type="checkbox"/> method names <input type="checkbox"/> indentation rules <input type="checkbox"/> Using inline comments frequently <input type="checkbox"/> Javadoc comments for every function (except getters/setters) 	Overall OOP design and API usage: <ul style="list-style-type: none"> <input type="checkbox"/> Appropriate use of inheritance <input type="checkbox"/> Appropriate use of Abstract classes <input type="checkbox"/> Use of Android API classes/methods (Not in base code) <input type="checkbox"/> Greater than 5 original classes 	0-10	
Functionalities: <ul style="list-style-type: none"> <input type="checkbox"/> On-screen menus <input type="checkbox"/> Scores <input type="checkbox"/> Controllable character <input type="checkbox"/> Sensor interaction (touch/accelerometer/etc) <input type="checkbox"/> Level <input type="checkbox"/> Opponents (AI) <input type="checkbox"/> 3 or More Levels <ul style="list-style-type: none"> <input type="checkbox"/> Standardised data structures (2 marks) <input type="checkbox"/> Randomly/Procedurally Generated (2 marks) <input type="checkbox"/> Stored online (2 marks) 	Design quality (both game and other game screens): <ul style="list-style-type: none"> <input type="checkbox"/> Professional looking <input type="checkbox"/> Understandable Game flow <ul style="list-style-type: none"> <input type="checkbox"/> Does not Crash <input type="checkbox"/> Feedback to user instead of crashing, or recover <input type="checkbox"/> Runs smoothly without interruptions <input type="checkbox"/> Installs without error <input type="checkbox"/> Starts/exits without error 	0-20	
Improvements marks: <ul style="list-style-type: none"> <input type="checkbox"/> Online high score list <ul style="list-style-type: none"> <input type="checkbox"/> Works (5 marks) <input type="checkbox"/> Attempted (3 marks) <input type="checkbox"/> Multithreading improvements <ul style="list-style-type: none"> <input type="checkbox"/> Works (5 marks) <input type="checkbox"/> Attempted (3 marks) <input type="checkbox"/> User Level Creation <ul style="list-style-type: none"> <input type="checkbox"/> Works (5 marks) <input type="checkbox"/> Attempted (3 marks) <p>The report MUST discuss the implementation.</p>	<ul style="list-style-type: none"> <input type="checkbox"/> Other approved extension <ul style="list-style-type: none"> <input type="checkbox"/> Works (5 marks) <input type="checkbox"/> Attempted (3 marks) <p><i>Note either the attempted marks or the works marks will be given (so you can only receive 5 marks per attempt max, and there is a max of 10 marks for this section, but tick off any extra work done anyway!)</i></p> <p>For attempted marks to be awarded the feature must be at such a state, that the code could have worked but does not due to bugs or similar.</p>	0-10	
Other comments:			

Signed and dated (student):

Signed (demonstrator):

Receipt CS2JA16 CW 2 Demo	Student Name	Mark	Marker Signature
------------------------------	--------------	------	------------------